Anomaly Detection At The LHC Using Autoencoders: Technical Report

2325611s

November 6, 2021

Abstract

With the High Luminosity Large Hadron Collider (HL-LHC) upgrade set to produce 100 million more proton-to-proton collisions, there needs to be a fast and reliable method to filter for new physics signatures. This is the reason why the Large Hadron Collider (LHC) employs a trigger system, which consists of the L1-trigger system (L1T) and the High-Level trigger (HLT); this work will focus on the L1T, which is set to employ autoencoders (AEs) on Field Programmable Gate Arrays (FPGAs) for microsecond period inferencing. The main goal of this work is to produce machine learning (ML) models to do anomaly detection (AD) in order to find new physics that is beyond the Standard Model (BSM). ML is used because it would be incredibly difficult to create an algorithm manually to perform AD and it would not be able to generalise as well comparatively. This report lays the groundwork for the knowledge required to understand this project as well as the related work and goals.

Contents

1 Introduction

At the CMS detector in the LHC at the European Council for Nuclear Research (CERN), 40 million proton-to-proton collisions occur every second. With the HL-LHC upgrade [3], the frequency of collisions is expected to increase to 140 MHz from 40 Mhz, which will result in a significant increase in the amount of collision data. To process all that data, the LHC makes use of a trigger system, which includes two main triggers: the L1T and the HLT. The L1T is used for fast processing using FPGAs, which creates the need for a method to quickly identify potential new physics accurately, and this is where autoencoders come in. Autoencoders [5] were designed to be used for data compression for the likes of images, videos, denoising, etc. [4], however, they can be used to detect anomalies in high-energy physics (HEP). Autoencoders can be used to compress collision data into a bottleneck (latent space representation) that has a lower dimensionality compared to the input data, and then decompress that bottleneck into what the autoencoder thinks is what the input looked like (the reconstructed features of the collision data). By applying this to known physics - data with Standard Model (SM) events - we can run the autoencoder on data with potential new physics that's beyond the Standard Model (BSM), which will result in a poor reconstruction. The worse the reconstruction the more likely it is that the data contains a BSM event. This allows the L1T to identify potential new physics to be studied further with algorithms at the HLT.

It has been shown that machine learning model architectures such as Deep Neural Networks (DNNs) [9], Graph Neural Networks (GNNs) [17, 18], and Convolutional Neural Networks [12] perform well when it comes to classifying and detecting anomalies with HEP data. In this work, CNN and GNN architectures will be created and trained using TensorFlow [1] and Keras [8]. All models will be quantised (reducing the number of bits per weight) and pruned (reducing the number of weights) in order to reduce the model sizes, which is necessary due to the memory and time complexity constraints of the FPGAs in the L1T. Using *hls4ml* [9, 11] these models will be converted into these models cannot be complex, and therefore the number of layers (and weights) any given model can have is limited.

The rest of the report is organised as follows: Section 2 discusses the design of the LHC, in Section 3 the HL-LHC upgrade is detailed, Section 4 explains how the L1T trigger works at the CMS experiment, Section 6 elaborates on how the process of anomaly detection at the LHC works, Section 5 describes autoencoders and some commonly used autoencoder architectures, Section 7 goes over the AD work done before this project, and finally Section 8 summarises this report.

2 The Large Hadron Collider

The LHC [10] at CERN is the world's largest particle accelerator; with a circumference of 27km and a per beam energy of 6.5 TeV, which produces collisions at an energy of 13 TeV. The LHC is a large international collaboration between many institutes and organisations, which includes the School of Physics' Particle Physics Experiment group (PPE). The LHC was used to discover the long theorised Higgs Boson back in 2012. It has verified the predictions of the SM at very high energy scales and incredible precision in terms of the strong and weak forces. The LHC accelerates two particles beams, that are kept in their respective ultra-high vacuums, as close to the speed of light as possible in order to produce high-energy collisions. Trillions of particles are sent around the LHC at a frequency of over 10kHz, these particles are first sped up in a smaller linear tube where electromagnetic fields constantly push them forwards; then these particles enter a loop (much smaller than the LHC) where the particles are sped up even more before being injected into the 27km long ring of the LHC. The LHC contains a strong magnetic field that is produced by superconducting electromagnets (in total there are 50 different types of magnets used), these electromagnets operate at 1.9K, are cooled by liquid helium, and their main dipoles generate magnetic fields with a magnitude of 8.3T and a current of 11kA.

Shown in Figure 1 and 2 are the schematic views of the LHC and the 4 main experiements - CMS, ATLAS, LHCb, and ALICE. These experiments are where the two particle beams collide, and so detectors (that each house over 100 million sensors) are placed in order to record the aftermath of the collisions. The particles beams typically consist of protons, which themselves are made up of quarks and gluons that hold the quarks together via the strong force. Usually, when two protons collide ordinarily, they

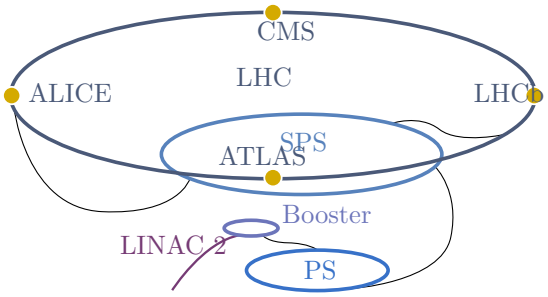

Figure 1: A diagram of the main elements of the LHC, which includes the main ring, the four experiments in yellow, as well as the smaller accelerators used to speed up the particles to near the speed of light [13].

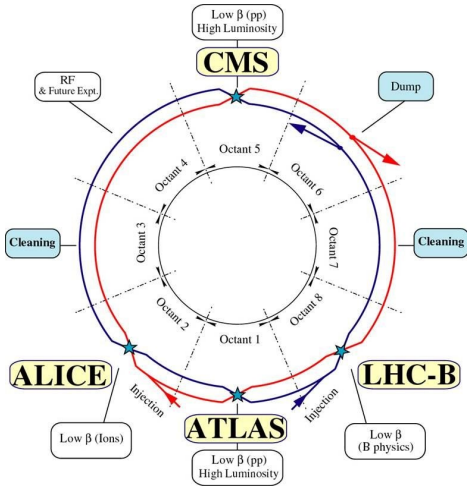

Figure 2: Schematic view of the beams and how they cross at the four main experiments of the LHC [10].

pass through each other without producing anything of note. However, with the energy the LHC gets these two beams up to, there's a one in a million chance that two protons collide and the collision energy is set free, creating thousands of particles.

The LHC operates by doing "runs", where the detector is turned on and the collisions occur, followed by a prolonged period of rest where the data is analysed to detect new physics signatures. So far, two of these runs have been completed, and the third run has already started and is set to produce even more data than before. The very first run, which was the run that produced the Higgs Boson, ran as high as 4TeV per beam. The second run produced beams with 7.5TeV (collision energy of 13TeV). The third run is set to produce collisions with an energy of 14TeV.

3 High Luminosity Large Hadron Collider

The current limitation of the LHC is both the energy it is able to collide these protons with (13TeV) and the frequency with which the collisions occur. There was a hope that run 2 would produce evidence for SUSY particles at around the energy limit of what the LHC is able to do, this, unfortunately, turned out to be not the case. However, with alterations to the SUSY theory, it is possible that they actually are more massive than was originally thought. Therefore there is a need for the LHC to produce more high-energy collisions than before. Not only that but the number of collisions also needs to increase, as detecting particles like the Higgs Boson requires a lot of collision data. The odds of producing the Higgs Boson for example is one in 10 billion. Since the detector does not poll quick enough to detect the Higgs Boson, its existence needs to be determined by what it decays into, which is sometimes two high-energy photons. However, there are many other particles that decay in such a way, so instead the mass of the Higgs Boson needs to be figured out by working backwards. In order to do this, all events that produce two photons are graphed and only until a statistically significant bump appears on top of the background where the mass of the Higgs Boson is theorised to be, that it can be said that it was detected. This graph is shown in Figure 3.

This process takes a very long time (billions of observations) due to the unlikely nature of producing these particles that are very short-lived. The HL-LHC upgrade aims to increase the frequency of collisions from 40Mhz to 140Mhz in order to make these events more frequent, so we can study not only the Higgs Boson but other short-lived particles better too. The HL-LHC upgrade is set to collide protons at a centre-of-mass ($\sqrt{s}$) of 14 TeV, and increase the integrated luminosity to $3ab^{-1}$.

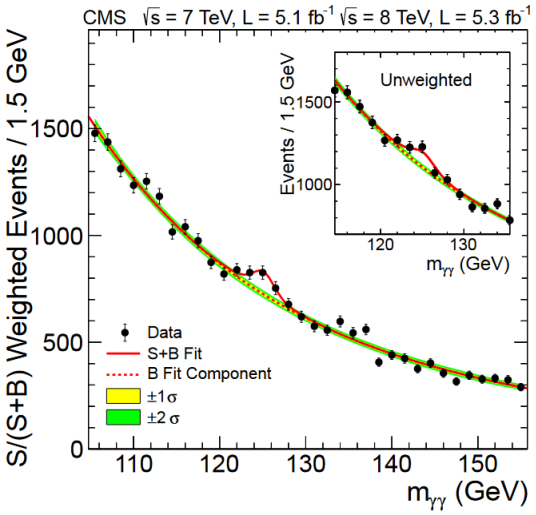

Figure 3: The diphoton invariant mass distribution graph showing the Higgs Boson at a mass of 125GeV, produced during run 1 [7].

4 CMS Trigger System

One of the challenges the CMS experiment faces is the amount of data that needs to be processed from the collisions - it is dubbed as the 40Mhz challenge. With trillions of particles being sent around the LHC, 40 million collisions occur every second or in other words there are 25 ns until the next set arrives, which produces hundreds of terabytes of data to be processed every second. With the HL-LHC upgrade, the frequency of collisions is expected to increase to 140 MHz, increasing the data produced every second by the LHC to exabytes of collision data every second. It is not feasible to store all of this data, and therefore the collisions need to be filtered. Only around 1000 events per second can be stored, which is a significant reduction in the amount of data being stored compared to the amount being produced. When the two proton beams collide, the detectors capture the resulting jet streams and filter them through a real-time processing system called a trigger [15]. There are two parts to the trigger system, firstly the data is processed by the L1T which houses FPGAs for microsecond period processing (shown in Figure 4) and then is moved to the HLT for longer, more complex, analysis and filtering.

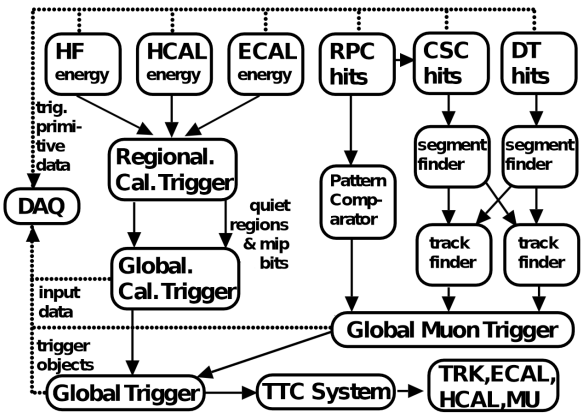

Figure 4: The diagram for the CMS L1T [15]. Data from the forward (HF) and barrel (HCAL) hadronic calorimeters, and from the electromagnetic calorimeter (ECAL), are processed first regionally (RCT) and then globally (GCT). Energy deposits (hits) from the resistive- plate chambers (RPC), cathode strip chambers (CSC), and drift tubes (DT) are processed either via a pattern comparator or via a system of segment- and track-finders and sent onwards to a global muon trigger (GMT). The information from the GCT and GMT is combined in a global trigger (GT), which makes the final trigger decision. This decision is sent to the tracker (TRK), ECAL, HCAL or muon systems (MU) via the trigger, timing and control (TTC) system. The data acquisition system (DAQ) reads data from various subsystems for offline storage. MIP stands for minimum-ionizing particle.

This work will concentrate around the L1T exclusively, so the processing constraints of the L1T will need to be taken into account when creating the models. the L1T takes in 40MHz of data and filters that down to 100KHz, which is fed into the HLT where it is finally reduced down to 1KHz. The processing done at the L1T needs to happen in the order of microseconds, while at the HLT can take 100s of milliseconds to process data further. The CMS trigger system was chosen due to my involvement with the CMS experiment as part of the CERN Summer Student programme, which included doing AD research related to the triggers. This work will be a continuation of that research, which is why ATLAS' L1T is not being looked at when Glasgow has an ATLAS group within PPE, but not a CMS group.

5 Autoencoders

AEs use unsupervised machine learning, where the ground truth of the training dataset is not known, to do representation learning. The idea is to impose a bottleneck inside the network in order to compress the representation of the knowledge the network has learned from the input dataset. AEs were originally created for the purpose of using machine learning to create a more space and time-efficient method of compressing and uncompressing image data, although they did not excel in this regard, they have great use cases in many other domains. An AE consists of three main parts (shown in Figure 5), the encoder, the latent space, and the decoder. The encoder takes in the input data (usually a tensor) and uses many layers to compress the dimensionality of the original dataset into a much smaller representation. This compressed representation of the original input data is called the latent space and is typically one or two dimensional in order to save on both the space it takes up and the computation power required to process it. This latent space representation of the input data is then input into the decoder, where the reverse of what the encoder has done happens. The decoder tries to decompress the latent space back into what it thinks is the input data, this is called the reconstructed data. Typically, the architecture of the decoder is more complex as it is harder to decode data than it is to simply encode it into a smaller space representation, although the encoder's importance cannot be neglected as it is the encoder that is often the downfall to many AE architectures.

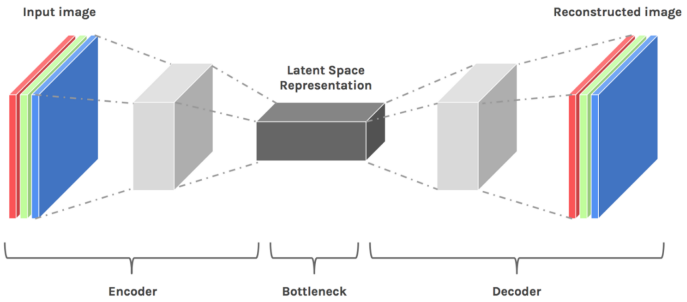

Figure 5: A digram of the basic structure of an AE, with the encoder compressing down to the latent space and the decoder learning to reconstruct the original image input [16]. The light grey blocks represent the many possible hidden layers.

There are many ways to implement AEs, that is to say, the architecture of the layers themselves and how the weights and biases in each layer interact with the previous and next layers. Often a Convolutional Neural Network (CNN) architecture is used to create an AE, this is because of the ability of CNNs to learn features of the dataset effectively. Depending on the kernels used, the autoencoder could be set up to pick out the most important features of the dataset, and this applies not only to images but generally to three-dimensional data too. In a CNN AE, the encoder features regular convolutional layers that use kernels (or sometimes referred to as filters) to compress the image while biasing towards certain parts of the input data, while the decoder features deconvolutional layers to do the opposite. The general expression for a convolution is shown in Equation 1.

$$c(x,y) = k * i(x,y) = \sum_{dx=-a}^{a} \sum_{dy=-b}^{b} k(dx,dy) * i(x+dx, y+dy) \tag{1}$$

Where c is the convoluted image, i is the input, and k is the kernel.

Graph Neural Networks (GNNs) have seen success in high-energy physics [2, 17] and in use as the layers for AEs used for anomaly detection at the LHC. For example, this paper [17] saw great performance with CMS data with a GNN based AE. Graph autoencoders (GAEs) do well under the constraints of unsupervised learning. They work by encoding the node features and the construction of the graph into the latent space and then decoding the latent space into the graph construction. To learn the space representation in the form of vectors, they use graph convolutional network (GCN) layers. GAEs also require the need for an adjacency matrix as the second input (the first being the background data), which includes whether or not the different pairs of nodes are adjacent within the graph. Getting the adjacency matrix right is critical to good model performance. AEs with a GNN layer structure are considered to be at the limit of what the L1T's hardware can handle.

6 Anomaly Detection

AD is the process of detecting new physics signatures from the collision data produced by the experiments at the LHC. AD is difficult to do manually, so this process has to be automated due to the strict constraint of the L1T to process the data in milliseconds and the complexity of the data itself. An algorithm, created manually, for such a task would also not be feasible as there are thousands of complex relations to consider. Previously, the time-consuming process of AD only happened when something went wrong at one of the LHC's experiments. This leads to the use of machine learning (ML) for such a task since it is great at learning large amounts of complex relationships by itself - without the need for human intervention once the ML models have been trained. ML works by training itself on the data with ground truth labels to tell the model what its guess should have been for the given data, this is called supervised learning. However, to ensure the ML model is learning to detect anomalies and not things that are already known, an unsupervised approach to learning has to be used, which is to say that during training it is not given those ground truth labels and instead will learn to understand the structure of the data itself. Unsupervised learning training is a much more complex undertaking than supervised learning as more work has to be done in order to get great results while not falling into the trap of overfitting the model to the data.

Many types of ML architectures can be used, with AEs [5] being the most prevalent in the field of particle physics AD. The way AEs are used to detect anomalies is that they are first trained on the background data that is comprised entirely of SM events. It learns the most important features of the background data, as the goal of the AE is to minimise the loss produced. Loss is the function used to tell the AE how far away it was from the data that it was trying to represent, both the encoder and decoder have a loss step in order to train for not only the reconstructed image but the encoded image that's in the latent space. Typically, the loss function is averaged, and so a popular choice is the mean squared error (MSE) function. MSE is shown in Equation 2, where n is the number of predictions, Y_i is the predicted value, and $\hat{Y}_i$ is the input value.

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(Y_i - \hat{Y}_i)^2 \tag{2}$$

In order to make use of the latent space efficiently, the AE needs to extract the relationships in the input data. Since the AE was trained on the SM events, anything beyond the events from the SM (BSM) is not going to be encoded or decoded properly and so there will be a major mismatch between the input event and the reconstructed event (the loss for the anomalous event will be larger than the background) - this, in essence, is how AEs are used for AD. To correctly do AD, the background training data needs to include as many SM events as possible. Even if a given SM event is not present in the background, the AE should be able to recognise it as such since it learned the features of SM events; this is the main advantage of the unsupervised approach. The loss function is used as the anomaly score to gauge how well the model performs.

7 Related Work

One of the first papers to document the use of an AE for AD at the LHC was [6], which used a Deep Neural Network (DNN) variational AE to show that not only is it possible to use an AE for such a task but it also performs well too - even when the model itself is compressed to fit on an FPGA. The model used there was able to achieve an AUC (Area Under the receiver operating characteristic Curve) of 92% in the best case and 75% in the worst case. That paper was followed by a paper that explored GNNs for use in particle physics AD [18], which achieved an AUC score of 98% even when compressed (quantised with Qkeras [14]). This was the first paper to explore a GNN based AE to detect new physics signatures at the LHC, the custom graph implementation was named GarNet. The way GarNet works is it builds the edges of the graph with each vertex in the graph collecting the information about the features across the edges. The results were important as it showed that it's possible to compress the model using *hls4ml* [9, 11] with very minimal loss in performance - an AUC reduction of 2% with a bit size reduction of a half. Both of these models were implemented using TensorFlow [1] and Keras [8]. CNNs were later explored in both variational and non-variational forms [12], where the results were also good (95% AUC). This paper also improved the performance of DNNs, and the models from this paper will later be deployed in CMS' L1T for run 3.

8 Summary

To summarise, a two-level trigger system is used at the LHC to process the vast amounts of collision data being produced by the detectors. The HL-LHC upgrade requires the need for even faster processing of data since it increases the frequency of collisions drastically. For this project, CNN and GNN autoencoder architectures will be investigated to see how well they detect anomalies on the much larger HL-LHC dataset, and how much of their respective performance they keep after the models are compressed using quantisation and pruning. The compression is required in order to fit the models on the custom hardware present in the first part of the trigger system (L1T), and by reducing their size, they will also be able to detect anomalies faster.

References

[1] Abadi, M., et al. Tensorflow: A system for large-scale machine learning, 2016.

[2] Abdughani, M., Ren, J., Wu, L., and Yang, J. M. Probing stop pair production at the lhc with graph neural networks. *Journal of High Energy Physics 2019*, 8 (Aug 2019).

[3] Aberle, O., et al. *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report.* CERN Yellow Reports: Monographs. CERN, Geneva, 2020.

[4] Alexandre, D., Chang, C.-P., Peng, W.-H., and Hang, H.-M. An autoencoder-based learned image compressor: Description of challenge proposal by nctu, 2019.

[5] Bank, D., Koenigstein, N., and Giryes, R. Autoencoders, 2021.

[6] Cerri, O., et al. Variational autoencoders for new physics mining at the large hadron collider. *Journal of High Energy Physics 2019*, 5 (May 2019).

[7] Chatrchyan, S., et al. Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc. *Physics Letters B 716*, 1 (Sep 2012), 30–61.

[8] Chollet, F. keras. `https://github.com/fchollet/keras`, 2015.

[9] Duarte, J., et al. Fast inference of deep neural networks in fpgas for particle physics. *Journal of Instrumentation 13*, 07 (Jul 2018), P07027–P07027.

[10] Evans, L., and Bryant, P. LHC machine. S08001–S08001.

[11] Fahim, F., et al. hls4ml: An open-source codesign workflow to empower scientific low-power machine learning devices, 2021.

[12] Govorkova, E., et al. Autoencoders on fpgas for real-time, unsupervised new physics detection at 40 mhz at the large hadron collider, 2021.

[13] Gude, A. Measurement of the phistar distribution of z bosons decaying to electron pairs with the cms experiment at a center-of-mass energy of 8 tev.

[14] Jr., C. N. C., et al. Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors, 2021.

[15] Khachatryan, V., et al. The CMS trigger system. *Journal of Instrumentation 12*, 1 (Jan. 2017), P01020.

[16] Larrue, T., Meng, X., and Han, C. Denoising videos with convolutional autoencoders a comparison of autoencoder architectures.

[17] Qasim, S. R., Kieseler, J., Iiyama, Y., and Pierini, M. Learning representations of irregular particle-detector geometry with distance-weighted graph networks. *The European Physical Journal C 79*, 7 (Jul 2019).

[18] Yutaro, I., et al. Distance-weighted graph neural networks on fpgas for real-time particle reconstruction in high energy physics. *Frontiers in Big Data 3* (Jan 2021).